

[illegible]

I HEREBY CERTIFY THAT THIS PAPER OR FEE IS BEING DEPOSITED WITH THE UNITED STATES POSTAL SERVICE "EXPRESS MAIL POST OFFICE TO ADDRESSEE" SERVICE UNDER 37 C.F.R. § 1.10 ON THE DATE INDICATED ABOVE AND IS ADDRESSED TO THE ASSISTANT COMMISSIONER FOR PATENTS, BOX PATENT APPLICATION, WASHINGTON, D.C. 20231

Robert C. Kowert/BAB  
Conley, Rose & Tayon, P.C.  
P.O. Box 398  
Austin, TX 78767-0398  
Ph: (512) 476-1400

## **BACKGROUND OF THE INVENTION**

### 1. Field of the Invention

5           This invention relates to computer system and, more particularly, to arbitrating a plurality of voltage requests in a computer system.

### 2. Description of the Related Art

10           As processors become faster and provide additional features, they tend to consume more power. Increased power consumption is undesirable for several reasons. For example, as computer use becomes more prevalent, the overall societal cost of powering all of these computers increases. On a more individual level, increased power consumption is also undesirable. In all computer applications, increased power  
15           consumption leads to increased operating costs. In battery-operated notebook computers, increased power usage decreases the life of the batteries.

            Another undesirable consequence of increased power consumption is increased heat generation. As computer systems and the devices within computer systems consume  
20           more power, they generate more heat. This additional heat makes heat-related problems such as system crashes, lockups, random reboots, memory errors and damage to the internal devices more likely. In order to minimize these disadvantages and heat-related problems, computer system designers have attempted to reduce power consumption. Some computer designers have tried to reduce power consumed by individual  
25           components such as processors. However, this has led to many disparate power specifications. For example, early processors operated at the same voltage, which was typically 5 volts. Now, as processors are designed to have ever-decreasing power consumption, new processors may require different voltages than their predecessors and competitors. Additionally, newer processors may use split rail designs, requiring two  
30           different voltage levels: one for the external or I/O voltage and another for the internal or

core voltage. For example, a split rail processor might require 3.3 V for its I/O voltage while requiring a lower core voltage of 2.9 V. In some systems, some components including the processor may operate at one voltage level, and other components, such as I/O components, may operate at another voltage level. Also, a manufacturer may offer  
5 different versions of a processor that operate at different voltage levels.

The various voltages that might be required by a device may be supplied by one or more power supplies. These power supplies might be voltage regulators such as linear or switching regulators. In a computer system, the motherboard may contain one or more  
10 voltage regulators. The voltage regulator may be configured to supply a certain voltage to a device. If a new device, such as a different version of a processor, that requires a different voltage is used in the motherboard, the voltage regulator must be adapted to provide the new voltage. In order to allow handling of disparate device power requirements, the motherboard may have jumpers that allow different voltages to be  
15 selected. However, jumpers present several disadvantages, the biggest of which is human error. If the person setting the jumpers has misidentified either the voltage needed or the correct jumpers to select, the wrong voltage will be generated and the device may not function properly. Additionally, jumpers may only facilitate a limited number of options. If no jumper setting can identify the necessary voltage, a new voltage regulator may be  
20 required in order to properly power the device.

As a result, some systems allow devices to request voltages from the voltage generator when the system starts up. For example, some systems use VID (voltage identification) codes instead of jumpers to select the voltages generated by the voltage  
25 regulators. At startup, the device outputs the VID code correlating to the voltage required. In this way, the device, not the user, controls what voltage it receives. This eliminates the possibility of erroneous voltage selection by the user and, as Table 1 shows, allows the identification of a large number of different voltages.

30 Table 1: Voltage Identification (VID) Codes

VRM Signal Name					Processor Core
VID[4]	VID[3]	VID[2]	VID[1]	VID[0]	Voltage (Vcc)
0	0	0	0	0	1.850V
0	0	0	0	1	1.825V
0	0	0	1	0	1.800V
0	0	0	1	1	1.775V
0	0	1	0	0	1.750V
0	0	1	0	1	1.725V
0	0	1	1	0	1.700V
0	0	1	1	1	1.675V
0	1	0	0	0	1.650V
0	1	0	0	1	1.625V
0	1	0	1	0	1.600V
0	1	0	1	1	1.575V
0	1	1	0	0	1.550V
0	1	1	0	1	1.525V
0	1	1	1	0	1.500V
0	1	1	1	1	1.475V
1	0	0	0	0	1.450V
1	0	0	0	1	1.425V
1	0	0	1	0	1.400V
1	0	0	1	1	1.375V
1	0	1	0	0	1.350V
1	0	1	0	1	1.325V
1	0	1	1	0	1.300V
1	0	1	1	1	1.275V
1	1	0	0	0	1.250V
1	1	0	0	1	1.225V
1	1	0	1	0	1.200V

1	1	0	1	1	1.175V
1	1	1	0	0	1.150V
1	1	1	0	1	1.125V
1	1	1	1	0	1.100V
1	1	1	1	1	Shutdown

Table 1 shows an example of 5-bit VID codes that a device might use to request a particular voltage from a power supply. Each of the 5 bits is sent as a signal to a voltage regulator module (VRM). The VRM is a voltage regulator that is programmed to accept a VID code and to output the voltage corresponding to the particular VID code received. For example, a VRM that received VID code 01011 would output a voltage of 1.575 V. Some computer systems may implement VID codes using more or less than 5 bits in order to accommodate a greater or lesser number of voltages. Additionally, different ranges of voltages might be represented using VID codes.

In computer systems that have several similar devices that perform similar functions, it is possible that some of the devices may request different voltages than others. For example, a multi-processor system may include two different processors and each processor may request a different voltage. In some systems, the devices may share a common voltage supply and consequentially, it may not be possible for each of the them to receive its requested voltage. In systems where each device has a dedicated power supply, if the devices each request different voltages from the power supplies, the power supplies will provide the devices with the different voltages requested. This creates a problem if the system configuration requires that the devices all operate at the same voltage. For example, some components that communicate with multiple processors may not be able to properly recognize signals that are input or output at different voltage levels. If the power supplies in the system supply each processor with its requested voltage, the processors may be operating at different voltage levels, causing their signals to have different voltage levels. This may cause the system to not function properly.

## SUMMARY

Various embodiments of a circuit and method for arbitrating voltage requests in a computer system are disclosed. In a first embodiment, one or more devices in a computer system are configured to each assert voltage request. These devices may comprise processors in some embodiments. An arbiter is configured to receive several voltage requests from the devices and to choose a voltage request to output to one or more power supplies. The power supplies are each configured to supply a voltage corresponding to the chosen voltage request to one or more of the devices. This allows all of the devices to receive the same voltage, even if they originally requested different voltages. In some embodiments, the arbiter may choose a voltage request from the voltage requests asserted by the devices. The different voltage requests may comprise VID codes. In some embodiments, the arbiter may include a programmable logic device configured to receive the voltage requests, choose a voltage request and output the chosen voltage request.

If only one device configured to assert a voltage request is present in the system, the arbiter may be configured to choose that device's voltage request. In some embodiments, the arbiter may be configured to receive a low power signal that indicates whether the devices should be in a low power state. The arbiter may be configured to choose a low power voltage request if the low power signal indicates that the devices should be in the low power state. In some embodiments, there may be multiple low power states, so the arbiter may also be configured to receive an additional low power signal. The arbiter may be configured to receive one or more power supply signals indicating whether any of the power supplies are not functioning properly. In certain embodiments, the arbiter may choose a low power voltage request if one of the power supply signals indicates that one of the power supplies is not functioning.

The arbiter may also be configured to receive several signals that indicate whether each of the one or more devices is actually present in the computer system in some embodiments. If none of the devices are present, the arbiter may be configured to choose

a low power voltage request. In other embodiments, the arbiter may be configured to choose a voltage request asserted by a device if the signals indicate that device is the only one of the devices present in the computer system.

- 5 Another embodiment discloses a method for arbitrating one or more voltage requests in a computer system. One or more voltage requests are asserted by one or more devices configured to receive a voltage. A voltage request is chosen. In some embodiments, the chosen voltage request is selected from the voltage requests asserted by the devices. The chosen voltage request is outputted to one or more voltage regulators.
- 10 The voltage regulators provide a chosen voltage corresponding to the chosen voltage request to the devices. In some embodiments, the devices include a processor. The voltage requests may include VID codes in some embodiments.

- 15 In another embodiment, a voltage request arbiter is disclosed. The voltage request arbiter has an input stage configured to receive a plurality of voltage requests from a plurality of devices, an arbitration stage configured to choose a chosen voltage request, and an output stage configured to assert the chosen voltage to one or more voltage regulators configured to provide a voltage to one or more of the devices.

## **BRIEF DESCRIPTION OF THE DRAWINGS**

Figure 1 is a block diagram of a computer system;

5        Figure 2 is a block diagram of an arbiter;

Figure 3 is a block diagram of a computer system containing an arbiter;

10       Figure 4 is block diagram of another embodiment of an arbiter;

Figure 5 is a flowchart illustrating one embodiment of a method of arbitrating different voltage requests;

15       Figure 6 is a flowchart illustrating a different embodiment of a method of arbitrating different voltage requests;

Figure 7 is a block diagram of another embodiment of an arbiter; and

20       Figure 8 is a PAL configured according to one embodiment of the invention.

25       While the invention is susceptible to various modifications and alternative forms, specific embodiments thereof are shown by way of example in the drawings and will herein be described in detail. It should be understood, however, that the drawings and detailed description thereto are not intended to limit the invention to the particular form disclosed, but on the contrary, the intention is to cover all modifications, equivalents and alternatives falling within the spirit and scope of the present invention as defined by the appended claims.



## DETAILED DESCRIPTION OF EMBODIMENTS

Figure 1 shows a computer system 100. The computer system 100 has one or more processors 101 coupled by one or more processor buses (collectively referred to as processor bus 103) to the north bridge 105. The north bridge 105, also known as a system controller, may contain such devices as a memory controller, a Peripheral Component Interconnect (PCI) bus controller, and an Accelerated Graphics Port (AGP). Preferably, the north bridge 105 forwards data, address and control signals between the processor bus, the PCI bus, the AGP and the memory. In order for components such as the north bridge 105 to properly interface with the processors 101, the processors 101 may all need to operate at the same voltage. For example, north bridge 105 may not be able to properly recognize signals at different voltage levels from processors 101. Thus, north bridge 105 may be unable to interface with processors operating at different voltages. The north bridge 105 is coupled to main memory 115 by data and address signals 125. The PCI bus 104 couples the north bridge 105 to the south bridge 106. The south bridge 106, also known as a peripheral bus controller, may contain devices such as a PCI to Industry Standard Architecture (ISA) bridge, an Enhanced Integrated Device Electronics (EIDE) controller, and a Universal Serial Bus (USB) controller. The south bridge 106 may perform administrative functions such as interrupt management, clock/calendar/timer functions, configuration management, power supply control, and power-on signal sequencing. The PCI bus 104 may be connected to one or more PCI devices 102.

Each of the processors 101 may have a separate power supply 145, shown as PS1 through PSn. Each processor outputs its voltage request in the form of a VID code, shown as VID1 through VIDn. Some of the processors 101 may require different voltages than others. If the processors have separate power supplies and the power supplies output the requested voltages, components such as north bridge 105 may not operate properly. Alternately, some embodiments of computer system 100 might have a single power supply for all processors 101. If there is only one power supply and that

power supply can only provide a single voltage, it may be impossible to supply each processor with its requested voltage.

Figure 2 shows a block diagram of a voltage request arbiter 200. The voltage request arbiter receives voltage requests, shown as VR1-VRn, from several devices, such as processors in an input stage 210. Then, the voltage request arbiter chooses a voltage request in an arbitration stage 220 and outputs the chosen voltage request to the power supplies PS1-PSn in an output stage 230. In some embodiments, there may be only one power supply to which the arbiter outputs the request. In some embodiments, the arbiter may choose a voltage request from the voltage requests asserted by the devices. In other embodiments, the arbiter may be configured to choose a new voltage request based on the voltage requests received. In some embodiments, the devices that assert the different voltage requests may be configured to operate properly at higher voltages but not lower voltages. In these embodiments, the arbiter might be configured to choose the voltage request requesting the highest voltage from the asserted voltage requests. The arbiter is configured to output the chosen voltage request to one or more power supplies that provide a voltage to the devices. This causes all of the devices to receive the same voltage. Devices that are supplied with a higher voltage than requested may require additional cooling. Since all the power supplies powering the devices receive the same voltage request, they supply the same voltage to all of the devices. The devices' requests may take the form of VID codes corresponding to the particular voltage required.

Figure 3 shows the voltage request arbiter 200 in the computer system 100 from Figure 1. The arbiter intercepts the individual voltage requests from processors 101 and chooses a single voltage request to output to all of the power supplies 145, labeled PS1-PSn. Now, each power supply PS1-PSn outputs the same voltage V corresponding to the chosen voltage request. As a result, all of the processors 101 operate at the same voltage. This may enable components such as the north bridge 105 to interface with the processors 101 properly.

In some embodiments, a computer system might have power conservation features. If the computer system has one or more low power states, such as standby, sleep or suspend, the system may require that a lower voltage be supplied to various devices. For example, when the computer system is in a normal state, a processor might require a core voltage of 1.9 V. While in one of the low power states, the same processor might receive a core voltage of 1.3 V. Devices other than processors may also need to be provided with different voltages when in low power states. There may be various low power states, and each low power state may have a different associated voltage that needs to be provided to the affected device(s). In order to provide these lower voltages, the voltage regulator generating the core voltage may need to receive voltage requests identifying the low power voltages. When the system exits the low power state, the affected devices' original voltage requests may be reasserted.

Figure 4 shows one embodiment of arbiter 200 that is configured to function in a computer system having one or more low power states like standby, sleep or suspend. In addition to receiving the devices' voltage requests, the arbiter may also receive one or more low power signal indicating whether the system is in a low power state. In some embodiments, a low power signal may be provided by a component such as a south bridge. When the system is in a low power state, the arbiter may ignore the devices' voltage requests and instead choose a low power voltage request to output to the power supplies. When the system leaves the low power state, the arbiter again chooses a voltage request to output.

Figure 5 shows a flow chart according to one embodiment of the invention. Several voltage requests are received at 505. In 510, checking is done to see if the computer system is in a low power state. If so, a voltage request corresponding to the low power state may be chosen, as indicated at 530. For example, if in a low power state, a VID of 10110 that corresponds to a low power voltage of 1.3V might be chosen. If not in a low power state, a non-low voltage state voltage request may be chosen, as indicated at 520. This voltage request may be chosen from one of the voltage requests received in

some embodiments. In other embodiments, the chosen voltage request may be a different voltage request than those received.

In Figure 6, a flow chart illustrates another embodiment of the current invention.

5 Several voltage requests are received at 605. In some embodiments, signals that indicate whether the voltage regulators are operational and signals that indicate which of the devices are present may be received. If a voltage regulator is not operational, it may be unsafe for that voltage regulator to provide a device with its requested voltage. Therefore, a different voltage request than one requested may be output if a voltage

10 regulator is not operational. In 610, some embodiments may determine whether the voltage regulators are working properly. If they are not, a voltage request corresponding to the low power state may be chosen, shown at 650. In some embodiments, the voltage request corresponding to a low power state may be a shutdown voltage request.

15 If the voltage regulators are working, some embodiments may check to see whether the computer system is asserting a low power state, as shown at 620. If so, in step 650 the voltage request corresponding to the low power state is chosen. In 620, some embodiments may check the signals indicating which of the devices are present. If only one device is present, there is no need for arbitration and that device's voltage

20 request is chosen as indicated at 620 and 630. If more than one device is present, all of the devices' voltage requests may be used to choose a voltage request to output. In some embodiments, the chosen voltage request may be selected from the voltage requests received. For example, in some embodiments, the highest voltage request received may be chosen. Some embodiments may choose a low power voltage request if the signals

25 indicate that none of the devices are present. In step 655, the chosen voltage request is output to one or more power supplies.

Figure 7 shows another embodiment of arbiter 200. In one embodiment, the arbiter may be configured to be connected to a dual-processor chipset (not shown). The

30 arbiter may be configured to arbitrate the two possible voltage requests that may be

generated by the two processors. These voltage requests might be expressed as VID codes labeled VID0 and VID1. To arbitrate between the requests, the arbiter circuit 700 may select one of them. For example, in some embodiments, the processors may be able to operate at a higher voltage than they request but not at a lower voltage. In these  
5       embodiments, the arbiter might determine which VID code represents the highest voltage and selects that VID code using the 5-bit comparator 705. Thus, the comparator may choose the VID code of either VID0 or VID1 depending on which VID code requests the highest voltage. In other embodiments, the arbiter may instead choose a new voltage request that differs from those asserted by the processors.

10

Table 2: Truth Table for MUX 710

X	Y	Z
0	0	A
0	1	C
1	0	B
1	1	x

Using multiplexer (MUX) 710, the arbiter may determine which, if any, of the processors are present. Table 2 shows a truth table for MUX 710. The active-low signals  
15       CPU0\_Pres\_L and CPU1\_Pres\_L indicate whether the first and second processors are actually present in the computer system. These two signals may be input as control signals to MUX 710. MUX 710 has three other inputs: VID0 (B), VID1 (C) and the higher of VID0 and VID1 (A). If only one processor is present, there is no need to arbitrate between different voltage requests and the MUX may select that processor's  
20       VID code request. If both processors are present, the MUX may instead select the VID code output by the comparator 705.

Table 3: Truth Table for MUX 715

X	Z
0	A

1	B
---	---

The arbiter may use MUX 715 to determine whether one of the conditions that might require the arbiter to select the low power state VID code is present. First, the two active-low signals indicating the processors' presence may be ANDed together. If neither processor is present, the output of the AND gate 720 is 1. Then, OR gate 730 may receive as inputs the output of the AND gate 720; an active-low signal CPU\_Stop\_L, which indicates whether the system is in a low power state; and two signals indicating whether the voltage regulators powering the processors are operating properly, P0\_DC\_OK and P1\_DC\_OK. If a power supply is not functioning properly, it may be unsafe to have that power supply provide a high voltage to a processor. Therefore, the arbiter may choose to assert the low power state voltage request if one of the power supplies is not working properly. In other embodiments, the arbiter may instead assert a shutdown voltage request if the power supplies are not operational. In some embodiments, if neither CPU is present, if the system has requested that the processors be put in a low power state, or if one of the power supplies is not functioning properly, OR gate 730 may output a 1. As shown in Table 3, this may cause the MUX 715 to select the VID code corresponding to the lower power state voltage, Stop\_VID, to output to the processors' power supplies.

If the system has not requested a low power state, at least one processor is present, and both of the processor power supplies are operating satisfactorily, the MUX 715 may output the VID code received from MUX 710. Thus, if more than one processor is present in the system, MUX 715 may output the VID code requesting the higher voltage to the voltage regulators powering the processors. Because the highest VID code is selected when both processors are present, the processor that requested a lower voltage may have to operate at a higher voltage. As a result, it may be desirable to have more cooling for this processor than it would require if it received the lower voltage it requested. In some embodiments of the invention, this additional cooling is provided. In some embodiments, selecting the lower voltage might result in the processor requesting

the higher voltage not functioning properly.

The functionality represented in Figures 2-7 might be implemented in many ways. For example, the above circuits might be implemented in a programmable logic device (PLD), such as a PLA (programmable logic array), field programmable gate array (FPGA) or PAL (programmable array logic), or a gate array or application specific integrated circuit (ASIC). Alternately, the arbiter might be implemented in another component of a computer system, such as a north bridge, south bridge, etc.

Figure 8 shows one embodiment of a VID arbiter implemented in a PAL. In this embodiment, the PAL used might be a PAL22V10 device. The PAL22V10 arbiter may be configured to arbitrate between two voltage requests from two processors, P0 and P1, in a computer system. Two active-low signals indicate whether either or both of two CPUs are present and are labeled CPU0\_PRES\_L and CPU1\_PRES\_L. In this embodiment, CPU0\_PRES\_L and CPU1\_PRES\_L are received at pins 21 and 1 of the PAL22V10. Two other signals, P0\_DC\_OK and P1\_DC\_OK, show whether either or both of the power supplies configured to provide voltages to the two processors are working. These signals are received at pins 23 and 22 respectively. An active low signal CPU\_STOP\_L indicates whether the computer system has requested that the processors be placed in a low power state and is received at pin 13. The voltage request VID\_P0[4..0] from the first processor P0 is received at pins 7-11. The second processor P1's voltage request VID\_P1[4..0] is received at pins 2-6. The voltage request VID[4..0] chosen by the PAL22V10 arbiter may be output from pins 14,15,16,17 and 20. Pins 18 and 19 may be temporary or intermediate outputs that are used to implement all of the necessary functionality that is required in the PAL22V10 arbiter.

Table 4: PAL code for Figure 8

PIN 12	GND
PIN 24	VCC

	PIN 6..2	VID_P1[4..0]	COMBINATORIAL; INPUT
	PIN 11..7	VID_P0[4..0]	COMBINATORIAL; INPUT
	PIN 22	P1_DC_OK	COMBINATORIAL; INPUT
	PIN 23	P0_DC_OK	COMBINATORIAL; INPUT
5	PIN 1	CPU1PRES_L	COMBINATORIAL; INPUT
	PIN 21	CPU0PRES_L	COMBINATORIAL; INPUT
	PIN 13	CPUSTOP_L	COMBINATORIAL; INPUT
	PIN 20,17,16,15,14	VID[4..0]	COMBINATORIAL; OUTPUT
	PIN 19	TEMP_OUT3	COMBINATORIAL
10	PIN 18	TEMP_OUT2	COMBINATORIAL

```

STRING OUT9 '(VID_P1[4]*VID_P0[4])'
STRING OUT8 '(/VID_P1[4]*VID_P0[4])'
STRING OUT7 '(OUT9 +(/OUT8*VID_P1[3]*VID_P0[3]))'
15 STRING OUT6 '(OUT8 +(/OUT9*VID_P1[3]*VID_P0[3]))'
   STRING OUT5 '(OUT7 +(/OUT6*VID_P1[2]*VID_P0[2]))'
   STRING OUT4 '(OUT6 +(/OUT7*VID_P1[2]*VID_P0[2]))'
   STRING OUT1 '(TEMP_OUT3 +(/TEMP_OUT2*VID_P1[0]*VID_P0[0]))'
   STRING OUT0 '(TEMP_OUT2 +(/TEMP_OUT3*VID_P1[0]*VID_P0[0]))'

```

20

EQUATIONS

```

TEMP_OUT2 = OUT4+(/OUT5*VID_P1[1]*VID_P0[1])
TEMP_OUT3 = OUT5+(/OUT4*VID_P1[1]*VID_P0[1])

```

25

CASE (P1\_DC\_OK,PO\_DC\_OK,CPUSTOP\_L,CPU0PRES\_L,CPU1PRES\_L)

BEGIN

#b11110:

BEGIN

30

VID[4..0] = VID\_P1[4..0]



```

        END
#b11101:
        BEGIN
        VID[4..0] = VID_P0[4..0]
5         END
#b11100:
        BEGIN
        IF (/OUT1*OUT0) THEN
            BEGIN
10             VID[4..0] = VID_P1[4..0]
            END
        ELSE
            BEGIN
            VID[4..0] = VID_P0[4..0]
15             END
        END
    OTHERWISE:
        BEGIN
            VID[4..0] = #b10110
20         END
    END
END

```

Table 4 shows one embodiment of the arbiter that might be implemented in a PAL such as that shown in Figure 8. The equations in Table 4 are written in PALASM (TM) (Programmable Array Logic Assembly Language, Monolithic Memories/Advanced Micro Devices, Inc.) convention. The notation + represents the logical OR operation, \* represents the logical AND operation, and / represents the NOT operation. The pin assignments shown correspond to the pins in Figure 8. The equations show how the PAL might operate. For example, when the variables controlling the case statement equal 11110, it means that both power supplies are operating properly (P0\_DC\_OK =

P1\_DC\_OK = 1), the system has not requested that the processors be in a low power state (CPU\_STOP\_L = 1), processor P1 is present (CPU1\_PRES\_L = 0) and processor P0 is not present (CPU0\_PRES\_L = 1). Since only one processor is present and no special circumstances such as a low power state or a non-operational power supply are present, the arbiter chooses the voltage request VID\_P1[4..0] corresponding to the processor P1 and outputs that voltage request as VID[4..0]. When the case statement variables equal 11101, processor P0 is present and processor P1 is not and the arbiter may choose the voltage request VID\_P0[4..0]. If both processors are present, both power supplies are operating properly, and the system is not requesting a low power state, the case variables are 11100 and the PAL arbiter may use the equations shown in Table 4 to select the higher of the two voltage requests VID0 and VID1. The selected voltage request may then be asserted as VID[4..0]. If none of the above circumstances are present, the PAL may be configured to output a voltage request corresponding to a low power state. For example, if one of the power supplies does not indicate that it is working properly, the PAL arbiter may be configured to output the voltage request VID 10110, which corresponds to the low power state voltage of 1.3V. In other embodiments, a PAL arbiter might be configured to output a voltage request VID corresponding to a shutdown state if one of the power supplies is not working properly.

Numerous variations and modifications will become apparent to those skilled in the art once the above disclosure is fully appreciated. For example, even though many of the above examples describe situations involving processors, similar situations might arise involving other devices as well. It is intended that the following claims be interpreted to embrace all such variations and modifications.

25